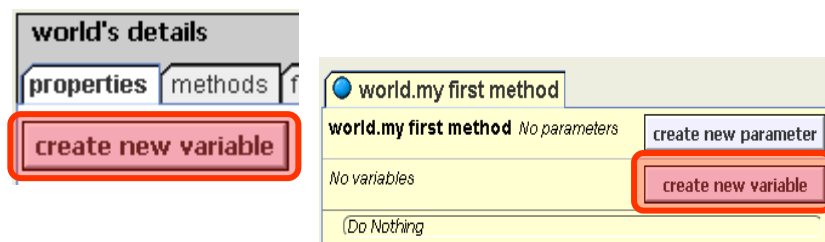
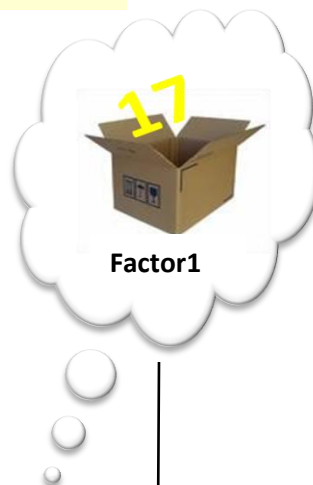
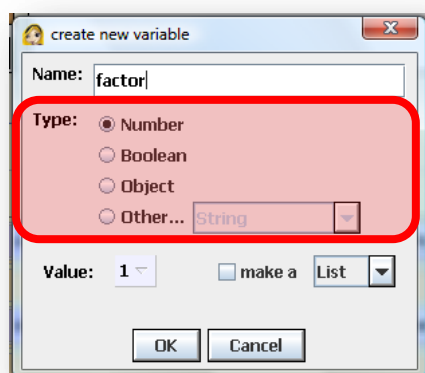


Variables

En Alice se puede crear una variable desde el panel de detalles *propiedades* de cualquier objeto o desde un método específico.



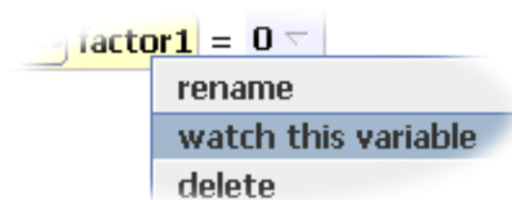
Las variables almacenan diferentes tipos de datos:



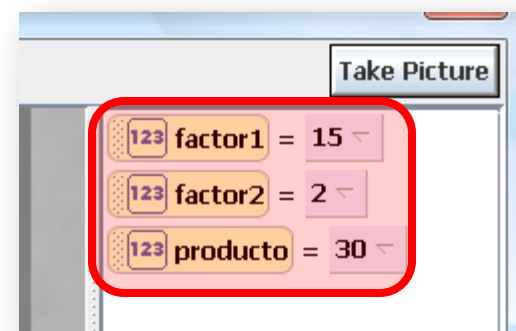
Por tanto una variable se compone de :



Se puede visualizar el comportamiento de una variable mientras el programa se encuentra en ejecución, únicamente da clic derecho sobre la variable y selecciona del menú **watch this variable**.

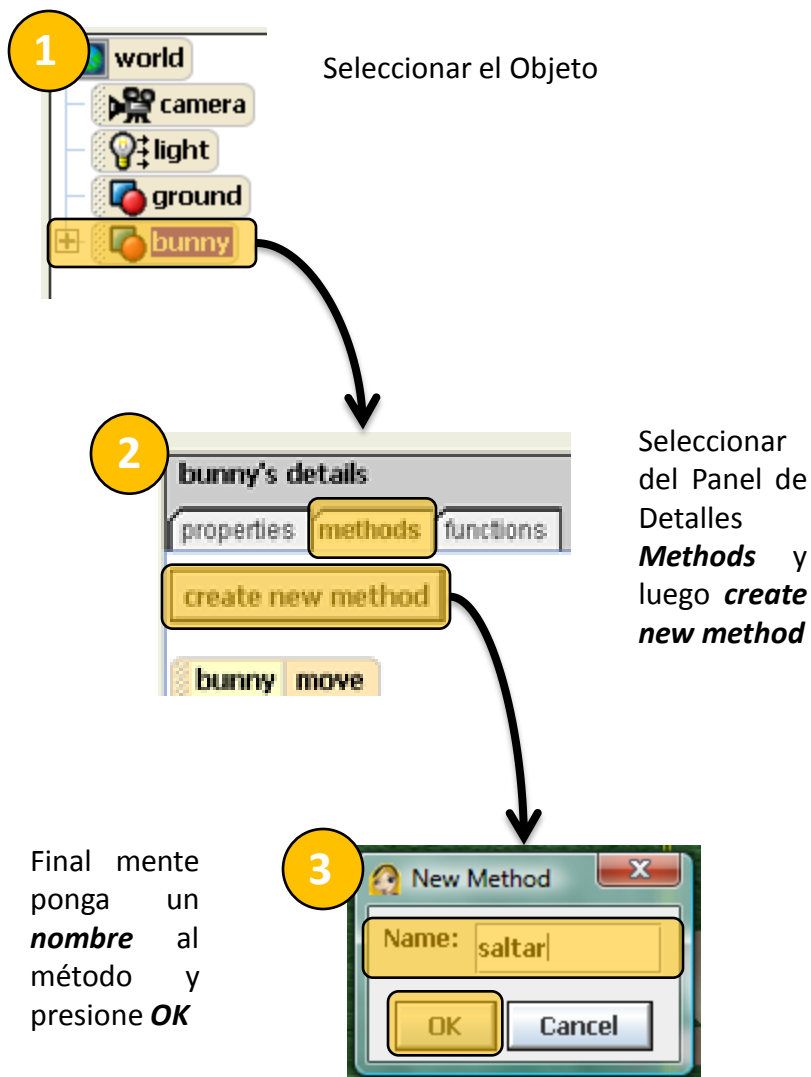


Esto permitirá observar al momento de ejecutar el mundo (**PLAY**) a la derecha, un panel con la variable, dónde se puede visualizar el comportamiento de la misma durante la ejecución del programa.



Taller 2

Cuando se construye un método se debe indicar el objeto al que pertenecerá:



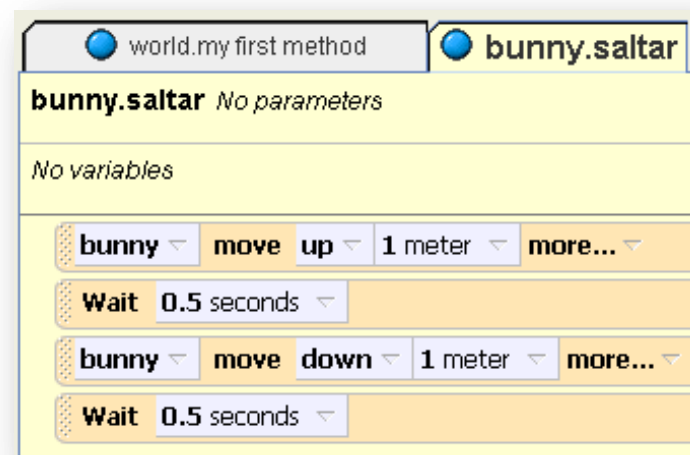
Trabajo con

Métodos

Lo anterior abrirá en el área de la derecha un nuevo bloque de programación.

El nombre del método está formado por:

Nombre del Objeto • nombre del método



En este ejemplo al Objeto bunny le crearon un método llamado saltar.

Importante: para que las instrucciones del método se ejecuten al momento de presionar el botón de Play, el método debe ser llamado desde otro método o bien en un evento. En la siguiente ficha se explica cómo hacerlo.

Taller 2

Hay dos maneras de invocar un método:

Para activar el método desde otro método,

Opción 1

Desde otro Método



Si desea ver el código del método, selecciona el botón Edit.

¿Qué son los Eventos?

Opción 2

Directamente en el evento

Para activar el método desde un evento,

A **create new event** (En caso que no dese trabajar con el evento por defecto*)

When the world starts , do world.my first method

When any key is typed , do Nothing C

Seleccionar en la casilla **nothing** el evento que desea ejecutar.

Evento	Acción
*When the World starts do	Se ejecuta al iniciar el mundo (Evento por defecto)
When a key is type	Se ejecuta al presionar una tecla
When the mouse is clicked on something	Se ejecuta al hacer clic en cualquier parte
While something is true	Se ejecuta mientras algo sea verdadero
When a variable changes	Se ejecuta al cambiar el contenido de una variable
Let the mouse move <objects>	Permite arrastrar el objeto con el mouse
Let the arrow keys move <subject>	Permite mover el objeto con las flechas del teclado
Let the mouse move the camera	Permite mover la cámara con el ratón
Let the mouse orient the camera	Permite orientar la cámara con el ratón

Al incluir un nuevo método aparecerán en algunos casos una, dos o más casillas con un texto en color rojo:

any key -> seleccione la tecla.

Nothing -> seleccione el método

any thing -> seleccione un objeto

none -> seleccione una condición

any object -> seleccione varios objetos de una lista

Taller 2

Una condicional es como venir por una calle y encontrarse una intersección (Y), depende para donde vayas, ó que quieras hacer, tomas uno de los dos caminos.

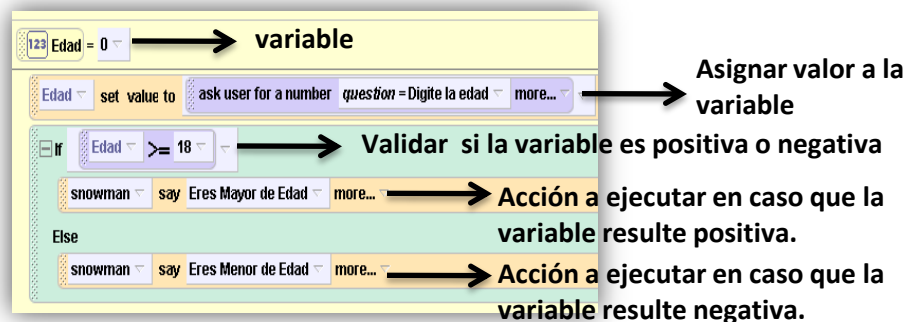
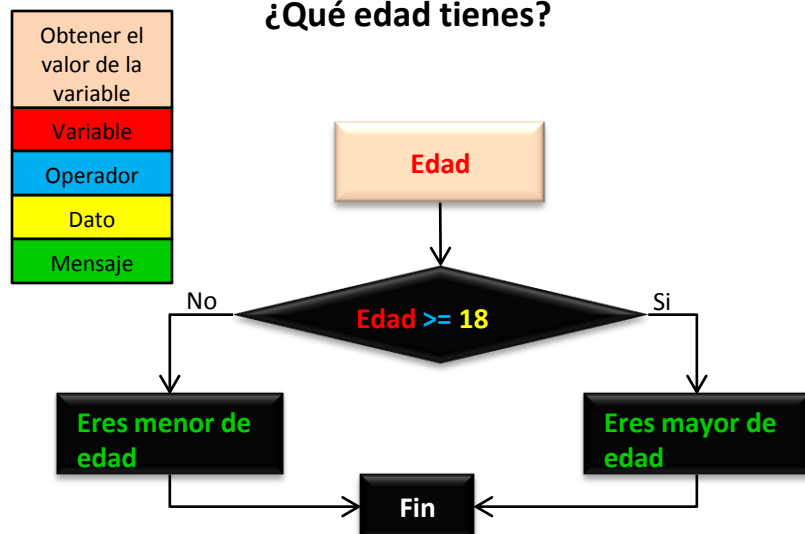
Para poder elaborar una condición se requiere la comparación de dos elementos (a y b) que bien pueden ser variables, datos u objetos, para realizar esta comparación se requiere de los operadores matemáticos ubicados en la sección **Funcios** del objeto **World**.

math

- $a == b$ → a y b deben ser iguales
- $a != b$ → a tiene que ser diferente de b
- $a > b$ → a tiene que ser mayor que b
- $a >= b$ → a tiene que ser mayor o igual que b
- $a < b$ → a tiene que ser menor que b
- $a <= b$ → a tiene que ser menor o igual que b

¿Qué son las Condicionales?

¿Qué edad tienes?



Do in order

Do together

If/Else

Loop

While

For all in order

For all together

Wait

print

///

¿Cómo hacer una Pausa?

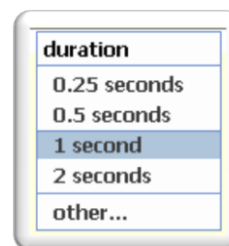
Cuando construimos programas en ocasiones es necesario hacer pausas en la ejecución del mismo, esto permitirá poder observar con detalle algún evento.

En realidad una pausa es como lo que ocurre con los semáforos que esperan cierta cantidad de segundos para cambiar de color, de lo contrario no habría tiempo para cruzar la calle.

Ejemplo del Semáforo



Al arrastrar la instrucción **Wait** aparece un menú del cual se escoge el tiempo de la pausa, si requiere un dato que no se encuentra en la lista seleccione la opción **Other...** (Otros...) y digite la cantidad de tiempo que requiere.



Do in order

Do together

If/Else

Loop

While

For all in order

For all together

Wait

print

///

Taller 2

Significa “**Bucle o ciclo de repetición**”

Esta sentencia permite realizar repetidas veces un bloque de instrucciones para no tener que escribir el mismo código “n” cantidad de veces al momento de programar.

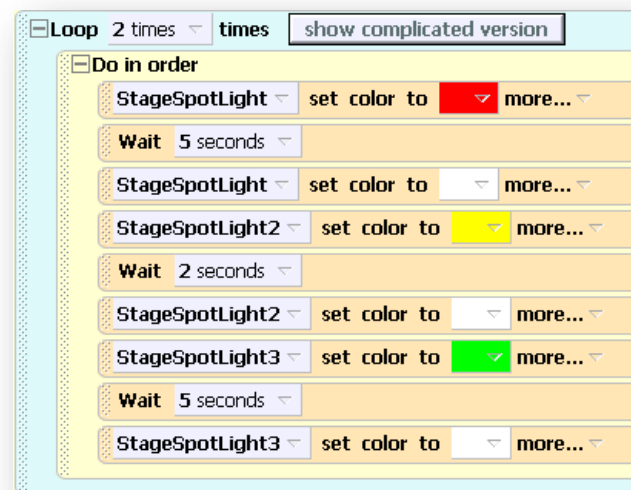
Si volvemos al ejemplo del semáforo podemos identificar el ciclo, cada vez que se enciende la luz verde se debe repetir el ciclo de luces nuevamente, esto lo hace indefinida mente.

Los ciclos se programan con un tiempo el cuál indica la cantidad de veces que se deben repetir las instrucciones, en ALICE se puede definir un ciclo como Infinito, esto quiere decir que mientras se ejecute el programa el ciclo nunca se detendrá.

¿Que es un LOOP?

Al definir un ciclo en Alice se puede delimitar las veces que se ejecute el ciclo, cuando arrastre la instrucción **Loop** aparecerá un menú del cual puede escoger la cantidad de veces que lo necesite o si es un ciclo infinito puede seleccionar *infinity times*

end
1 time
2 times
5 times
10 times
infinity times
other...



Do in order

Do together

If/Else

Loop

While

For all in order

For all together

Wait

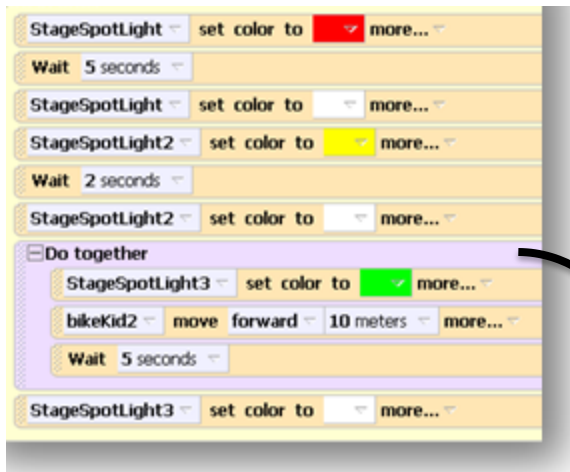
print



Taller 2

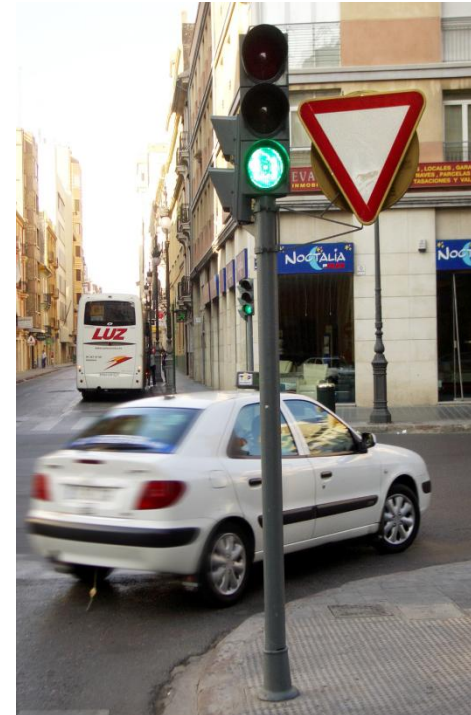
En ciertos momentos se necesitará que se ejecuten procesos de forma paralela (al mismo tiempo), todas las instrucciones que se coloquen dentro de esta instrucción se activarán al mismo tiempo.

En la vida real al colocarse el semáforo en verde los vehículos pueden avanzar, si se observa con atención se están realizando dos acciones simultáneamente, el semáforo cambia a color verde y los autos avanzan, por tal motivo estas dos acciones deben ejecutarse juntas. En ALICE para que esto se pueda realizar se puede hacer uso de la instrucción **Do Together**, que significa hacer juntos.



¿Que es un

Do Together?



Todas las instrucciones que se coloquen dentro del bloque de programación **Do together** se ejecuta simultáneamente, es por eso que al insertar dentro de este bloque la programación de la luz verde y el avance del ciclista obtenemos una animación de lo que sucede en realidad con la luz verde del semáforo.

Do in order

Do together

If/Else

Loop

While

For all in order

For all together

Wait

print

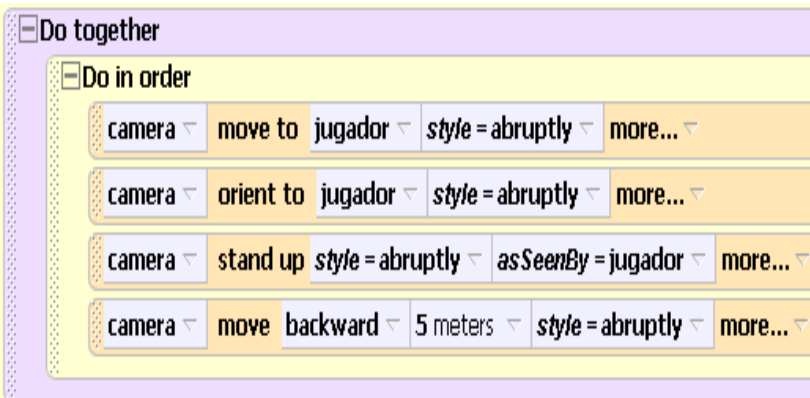
///

Do in order?

Significa “hacer en orden”

Muchas veces al incluir instrucciones dentro de un **Do Together** pareciera que no se ejecuta alguna de las instrucciones, esto se debe a que al ejecutarse al mismo tiempo no se logra ver el efecto.

El **Do in order** dentro del **Do together** permite ejecutar todas las instrucciones al mismo tiempo pero respetando un orden específico.



En este caso de programación se le esta pidiendo a la cámara que se desplace hacia un lugar específico mediante varias instrucciones, si estas instrucciones no se colocaran dentro del **do together** se vería un efecto pausado, pero el problema es que al ejecutarse todas al mismo tiempo no se obtiene el efecto que se pretendía, es por eso que se colocan dentro de un **do in Order**.

Do in order

Do together

If/Else

Loop

While

For all in order

For all together

Wait

print

///